Using Recognition to Guide a Robot's Attention

Using Recognition to Guide a Robot's Attention

Alexander Thomas*, Vittorio Ferrari†, Bastian Leibe‡, Tinne Tuytelaars* and Luc Van Gool‡
*Katholieke Universiteit Leuven, Belgium
Email: alexander.thomas/tinne.tuytelaars@esat.kuleuven.be
†University of Oxford, United Kingdom
Email: ferrari@robots.ox.ac.uk
‡ETH Zürich, Switzerland
Email: leibe/vangool@vision.ee.ethz.ch

***Abstract*— In the transition from industrial to service robotics, robots will have to deal with increasingly unpredictable and variable environments. We present a system that is able to recognize objects of a certain class in an image and to identify their parts for potential interactions. This is demonstrated for object instances that have never been observed during training, and under partial occlusion and against cluttered backgrounds. Our approach builds on the Implicit Shape Model of Leibe and Schiele, and extends it to couple recognition to the provision of meta-data useful for a task. Meta-data can for example consist of part labels or depth estimates. We present experimental results on wheelchairs and cars.**

I. Introduction

People are very strong at scene understanding. They quickly create a holistic interpretation of their environment. In comparison, a robot's interpretation comes piecemeal. A major difference lies in the human ability to recognize objects as instances of specific classes, and to feed such information back into lower layers of perception, thereby closing a *cognitive loop* (see Fig. 1). Such loops seem vital to 'make sense' of the world in the aforementioned, holistic way [14]. The brain brings all levels, from basic perception up to cognition, into unison. A similar endeavour in robotics would imply less emphasis on strictly quantitative – often 3D – modeling of the environment, and more on a qualitative analysis.

Indeed, it seems fair to say that nowadays robotics still has a certain preoccupation with gathering explicit 3D information (typically in the form of range maps) about the environment. Not only is this often a rather tedious affair, but many surface types defy 3D scanning altogether (e.g. dark, specular, or transparent surfaces may pose problems, depending on the scanner). Taking navigation as a case in point, it is known from human strategies that the image-based *recognition* of landmarks plays a far more important role than distance-based localisation with respect to some world coordinate system. The first such implementations for robot navigation have already been published [4, 3, 19]. This paper argues that modern visual object class recognition can provide useful cognitive feedback for many tasks in robotics[1].

The first examples of cognitive feedback in vision have already been implemented [9, 7]. However, so far they only coupled recognition and crude 3D scene information (the position of the groundplane). Here we set out to demonstrate the wider applicability of cognitive feedback, by inferring 'meta-data' such as material characteristics, the location and extent of object parts, or even 3D object shape, based on object class recognition. Given a set of annotated training images of a particular object class, we transfer these annotations to new images containing previously unseen object instances of the same class.

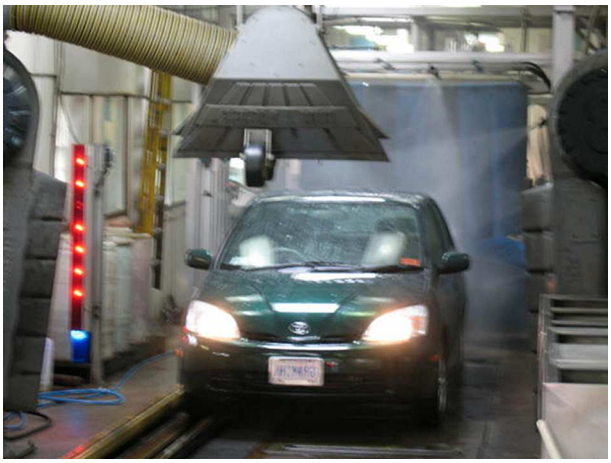

Fig. 1. Humans can very quickly analyze a scene from a single image. Recognizing subparts of an object helps to recognize the object as a whole, but recognizing the object in turn helps to gather more detailed information about its subparts. Knowledge about these parts can then be used to guide actions. For instance, in the context of a car wash, a decomposition of the car in its subparts can be used to apply optimized washing methods to the different parts.

There are a couple of recent approaches partially offering such inference for 3D shape from single images. Hoiem et al. [8] estimate the coarse geometric properties of a scene by learning appearance-based models of surfaces at various orientations. The method focuses purely on geometry estimation, without incorporating an object recognition process. It relies solely on the statistics of small image patches. In [20], Sudderth et al. combine recognition with coarse 3D reconstruction in a single image, by learning depth distributions for a specific type of scene from a set of stereo training images. In the same vein, Saxena et al. [18] are able to reconstruct coarse depth

[1]See also interview with Rodney Brooks in Charlie Rose 2004/12/21: http://www.youtube.com/watch?v=oEstOd8xyeQ, starting from 35:00

maps from a single image of an entire scene by means of a Markov Random Field. Han and Zhu [5] obtain quite detailed 3D models from a single image through graph representations, but their method is limited to specific classes. Hassner and Basri [6] infer 3D shape of an object in a single image from known 3D shapes of other members of the object's class. Their method is specific to 3D meta-data though, and their analysis is not integrated with the detection and recognition of the objects, as is ours. The object is assumed to be recognized and segmented beforehand. Rothganger et al. [15] are able to both recognize 3D objects and infer pose and detailed 3D data from a single image, but the method only works for object instances, not classes.

In this work, object related parameters and meta-data are inferred from a single image, given prior knowledge about these data for other members of the same object class. This annotation is intensely linked to the process of object recognition and segmentation. The variations within the class are taken account of, and the observed object can be quite different from any individual training example for its class. We collect pieces of annotation from different training images and merge them into a novel annotation mask that matches the underlying image data. Take the car wash scenario of Fig. 1 as an example. Our technique allows to identify the positions of the windshields, car body, wheels, license plate, headlights etc. This allows the parameters of the car wash line to better adapt to the specific car. Similarly, for the wheelchairs in Fig. 5, knowing where the handles are to be expected yields strong indications for a service robot how to get hold of the wheelchair.

The paper is organized as follows. First, we recapitulate the Implicit Shape Model of Leibe and Schiele [10] for simultaneous object recognition and segmentation (section II). Then follows the main contribution of this paper, as we explain how we transfer meta-data from training images to a previously unseen image (section III). We demonstrate the viability of our approach by transferring both object parts for wheelchairs and cars, as well as depth information for cars (section IV). Section V concludes the paper.

II. Object Class Detection with an Implicit Shape Model

In this section we briefly summarize the *Implicit Shape Model* (ISM) approach proposed by Leibe & Schiele [10], which we use as the object class detection technique underlying our approach (see also Fig. 2).

Given a training set containing images of several instances of a certain category (e.g. sideviews of cars) as well as their segmentations, the ISM approach builds a model that generalizes over within-class variability and scale. The modeling stage constructs a codebook of local appearances, i.e. of local structures that appear repeatedly on the training instances. Codebook entries are obtained by clustering image features sampled at interest point locations. Instead of searching for exact correspondences between a novel test image and model views, the ISM approach maps sampled image features onto

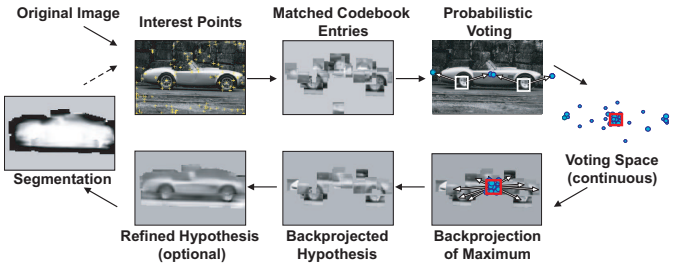

Fig. 2. The recognition procedure of the ISM system.

this codebook representation. We refer to the features in an image that are mapped onto a codebook entry as *occurrences* of that entry. The spatial intra-class variability is captured by modeling spatial occurrence distributions for each codebook entry. Those distributions are estimated by recording all locations where a codebook entry matches to the training images, relative to the annotated object centers. Together with each occurrence, the approach stores a local segmentation mask, which is later used to infer top-down segmentations.

A. ISM Recognition.

The ISM recognition procedure is formulated as a probabilistic extension of the Hough transform [10]. Let e be a sampled image patch observed at location ℓ. The probability that it matches to codebook entry c_i can be expressed as $p(c_i|e)$. Each matched codebook entry then casts votes for instances of the object category o_n at different locations and scales $\lambda = (\lambda_x, \lambda_y, \lambda_s)$ according to its spatial occurrence distribution $P(o_n, \lambda|c_i, \ell)$. Thus, the votes are weighted by $P(o_n, \lambda|c_i, \ell)p(c_i|e)$, and the total contribution of a patch to an object hypothesis (o_n, λ) is expressed by the following marginalization:

$$p(o_n, \lambda|e, \ell) = \sum_i P(o_n, \lambda|c_i, \ell)p(c_i|e) \qquad (1)$$

The votes are collected in a continuous 3D voting space (translation and scale). Maxima are found using Mean Shift Mode Estimation with a scale-adaptive uniform kernel [11]. Each local maximum in this voting space yields an hypothesis that an object instance appears in the image at a certain location and scale.

B. Top-Down Segmentation.

For each hypothesis, the ISM approach then computes a probabilistic top-down segmentation in order to determine the hypothesis' support in the image. This is achieved by backprojecting the contributing votes and using the stored local segmentation masks to infer the per-pixel probabilities that the pixel p is *figure* or *ground* given the hypothesis at location λ [10]. More precisely, the probability for a pixel p to be *figure* is computed as a weighted average over the segmentation masks of the occurrences of the codebook entries to which all features containing p are matched. The weights correspond

to the patches' respective contributions to the hypothesis at location x.

$$\begin{aligned} & p(p = figure|o_n, \lambda) \\ & = \sum_{p \in e} \sum_i p(p = figure|e, c_i, o_n, \lambda) p(e, c_i|o_n, \lambda) \\ & = \sum_{p \in e} \sum_i p(p = figure|c_i, o_n, \lambda) \frac{p(o_n, \lambda|c_i) p(c_i|e) p(e)}{p(o_n, \lambda)} \end{aligned} \tag{2}$$

We underline here that a separate local segmentation mask is kept for every occurrence of each codebook entry. Different occurrences of the same codebook entry in a test image will thus contribute different segmentations, based on their relative location with respect to the hypothesized object center.

In early versions of their work [10], Leibe and Schiele included an optional processing step, which refines the hypothesis by a guided search for additional matches (Fig. 2). This improves the quality of the segmentations, but at a high computational cost. Uniform sampling was used, which became untractable once scale-invariance was introduced into the system. We therefore implemented a more efficient refinement algorithm as explained in Section III-C.

C. MDL Verification.

In a last processing stage, the computed segmentations are exploited to refine the object detection scores, by taking only *figure* pixels into account. Besides, this last stage also disambiguates overlapping hypotheses. This is done by a hypothesis verification stage based on Minimum Description Length (MDL), which searches for the combination of hypotheses that together best explain the image. This step precludes, for instance, that the same local structure, e.g. a wheel-like structure, is assigned to multiple detections, e.g. multiple cars. For details, we again refer to [10, 11].

III. Transferring Meta-data

The power of the ISM approach lies in its ability to recognize novel object instances as approximate jigsaw puzzles built out of pieces from different training instances. In this paper, we follow the same spirit to achieve the new functionality of transferring meta-data to new test images.

Example meta-data is provided as annotations to the training images. Notice how segmentation masks can be considered as a special case of meta-data. Hence, we transfer meta-data with a mechanism inspired by that used above to segment objects in test images. The training meta-data annotations are attached to the occurrences of codebook entries, and transferred to a test image along with each matched feature that contributed to the final hypothesis (Fig. 3). This strategy allows us to generate novel annotations tailored to the new test image, while explicitly accommodating for the intra-class variability.

Unlike segmentations, which are always binary, meta-data annotations can be either binary (e.g. for delineating a particular object part or material type), discrete (e.g. for identifying *all* object parts), real-valued (e.g. depth values), or even vector-valued (e.g. surface orientations). We first explain how to transfer discrete meta-data (Section III-A), and then extend the method to the real- or vector-valued case (Section III-B).

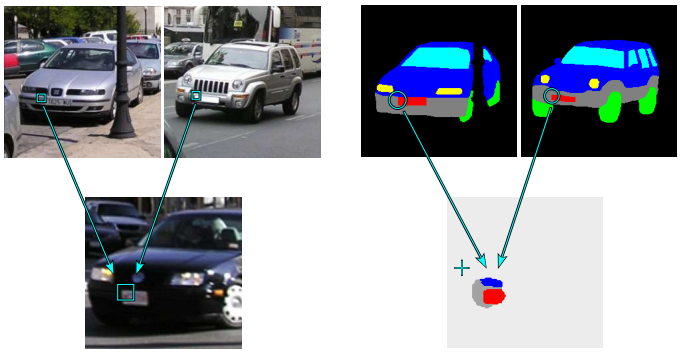

Fig. 3. Transferring (discrete) meta-data. Left: two training images and a test image. Right: the annotations for the training images, and the partial output annotation. The corner of the license plate matches with a codebook entry which has occurrences on similar locations in the training images. The annotation patches for those locations are combined and instantiated in the output annotation.

A. Transferring Discrete Meta-data

In case of discrete meta-data, the goal is to assign to each pixel of the detected object a label $a \in \{a_j\}_{j=1:N}$. We first compute the probability $p(p = a_j)$ for each label a_j separately. This is achieved in a way analogous to what is done in eq. (2) for $p(p = figure)$, but with some extensions necessary to adapt to the more general case of meta-data:

$$\begin{aligned} & p(p = a_j|o_n, \lambda) = \\ & \sum_{p \in N(e)} \sum_i p(p = a_j|c_i, o_n, \lambda) p\big(\hat{a}(p) = a_e(p)|e\big) p(e, c_i|o_n, \lambda) \end{aligned} \tag{3}$$

The components of this equation will be explained in detail next. The first and last factors are generalizations of their counterparts in eq. (2). They represent the annotations stored in the codebook, and the voting procedure respectively. One extension consists in transferring annotations also from image patches *near* the pixel p, and not only from those *containing* it. With the original version, it is often difficult to obtain full coverage of the object, especially when the number of training images is limited. This is an important feature, because producing the training annotations can be labour-intensive (e.g. for the depth estimates of the cars in Section IV-B). Our notion of proximity is defined relative to the size of the image patch e, and parameterized by a scalefactor s_N. More precisely, let an image patch e be defined by the three-dimensional coordinates of its center and scale e_λ obtained from the interest point detector, i.e. $e = (e_x, e_y, e_\lambda)$. The neighbourhood $N(e)$ of e is defined as

$$N(e) = \{p | p \in (e_x, e_y, s_N \cdot e_\lambda)\} \tag{4}$$

A potential disadvantage of the above procedure is that with $p = (p_x, p_y)$ outside the actual image patch, the transferred annotation gets less reliable. Indeed, the pixel may lie on an occluded image area, or small misalignment errors may get magnified. Moreover, some differences between the object instances shown in the training and test images that were not

noticeable at the local scale can now affect the results. To compensate for this, we add the second factor to eq. (3), which indicates how probable it is that the transferred annotation $a_e(p)$ still corresponds to the 'true' annotation $\hat{a}(p)$. This probability is modeled by a Gaussian, decaying smoothly with the distance from the center of the image patch e, and with variance related to the size of e by a scalefactor s_G:

$$\begin{aligned} p\big(\hat{a}(p) = a_e(p) \,|\, e\big) &= \frac{1}{\sigma\sqrt{2\pi}} exp\left(-({d_x}^2 + {d_y}^2)/(2\sigma^2)\right) \\ \text{with} \qquad \sigma &= s_G \cdot e_\lambda \\ (d_x, d_y) &= (p_x - e_x, p_y - e_y) \end{aligned} \tag{5}$$

Once we have computed the probabilities $p(p = a_j)$ for all possible labels $\{a_j\}_{j=1:N}$, we come to the actual assignment: we select the most likely label for each pixel. Note how for some applications, it might be better to keep the whole probability distribution $\{p(p = a_j)\}_{j=1:N}$ rather than a hard assignment, e.g. when feeding back the information as prior probabilities to low-level image processing.

An interesting possible extension is to enforce spatial continuity between labels of neighboring pixels, e.g. by relaxation or by representing the image pixels as a Markov Random Field. In our experiments (Section IV), we achieved good results already without enforcing spatial continuity.

B. Transferring Real- or Vector-valued Meta-data

In many cases, the meta-data is not discrete, but rather real-valued (e.g. 3D depth) or vector-valued (e.g. surface orientation). We can approximate these cases by using a large number of quantization steps and interpolating the final estimate. This allows to re-use most of the discrete-case system.

First, we discretize the annotations into a fixed set of 'value labels' (e.g. 'depth 1', 'depth 2', etc.). Then we proceed in a way analogous to eq. (3) to infer for each pixel a probability for each discrete value. In the second step, we select for each pixel the discrete value label with the highest probability, as before. Next, we refine the estimated value by fitting a parabola (a $(D+1)$-dimensional paraboloid in the case of vector valued meta-data) to the probability scores for the maximum value label and the two immediate neighbouring value labels. We then select the value corresponding to the maximum of the parabola. This is a similar method as used in interest point detectors (e.g. [12, 1]) to determine continuous scale coordinates and orientations from discrete values. Thanks to this interpolation procedure, we obtain real-valued annotations. In our 3D depth estimation experiments this makes a significant difference in the quality of the results (Section IV-B).

C. Refining Hypotheses

When large areas of the object are insufficiently covered by interest points, no meta-data can be assigned to these areas. Using a large value for s_N will only partially solve this problem, because there is a limit as to how far information from neighboring points can be reliably extrapolated. A better solution is to actively search for additional codebook matches in these areas. The refinement procedure in early versions

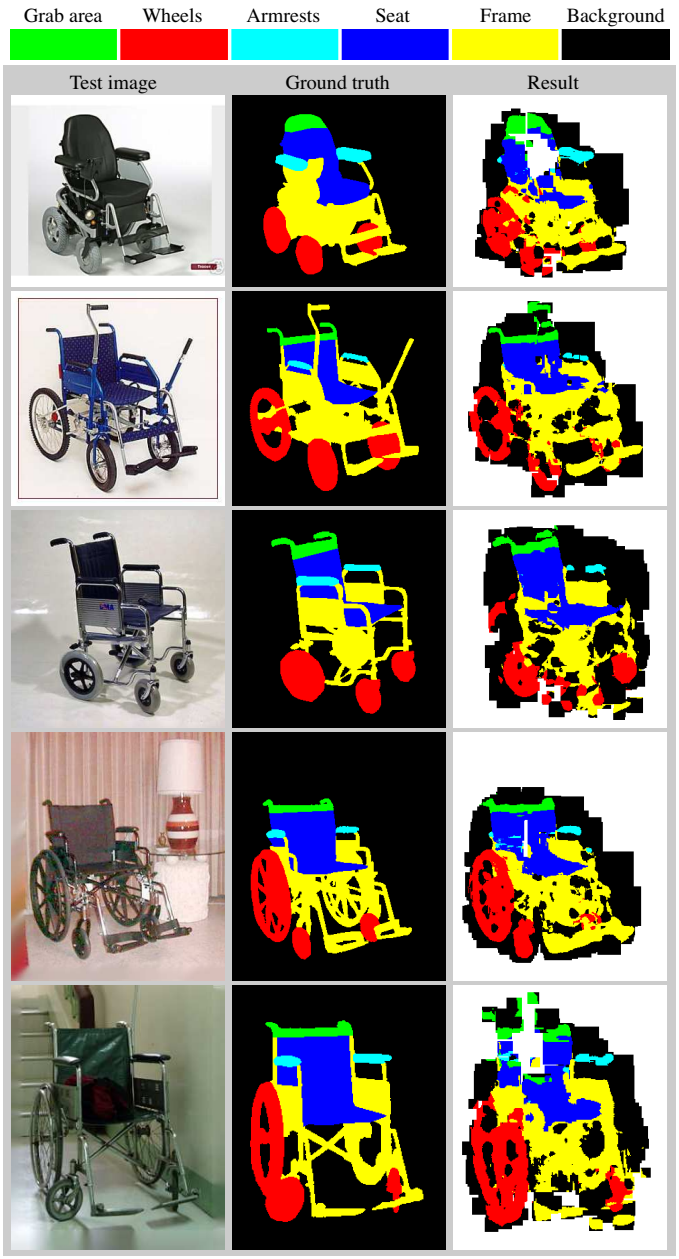

Fig. 4. Results for the annotation verification experiment on wheelchair images. From left to right: test image, ground-truth, and output of our system. White areas are unlabeled and can be considered background.

of the ISM system [10] achieved this by means of uniform sampling, which is untractable in the scale-invariant case. Therefore we implemented a more efficient refinement algorithm which only searches for matches in promising locations.

For each hypothesis, new candidate points are generated by backprojecting all occurrences in the codebook, excluding points nearby existing interest points. When the feature descriptor for a new point matches with the codebook cluster(s) that backprojected it, an additional hypothesis vote is cast. The confidence for this new vote is reduced by a penalty factor to reflect the fact that it was not generated by an actual interest

point. The additional votes enable the meta-data transfer to cover those areas that were initially missed by the interest point detector.

This refinement step can either be performed on the final hypotheses that result from the MDL verification, or on all hypotheses that result from the initial voting. In the latter case, it will improve MDL verification by enabling it to obtain better figure area estimates of each hypothesis [10, 11]. Therefore, we perform refinement on the initial hypotheses in all our experiments.

IV. Experimental evaluation

We evaluate our approach on two different object classes, wheelchairs and cars. For both classes, we demonstrate by means of a discrete labeling experiment, how our system simultaneously recognizes object instances and infers areas of interest. For the cars, we additionally perform an experiment where a 3D depth map is recovered from a single image of a previously unseen car, which is a real-valued labeling problem.

A. Wheelchairs: Indicating Areas of Interest for an Assistive Robot

In our first experiment, the goal is to indicate certain areas of interest on images of various types of wheelchairs. A possible application is an assistive robot, for retrieving a wheelchair, for instance in a hospital or to help a disabled person at home. In order to retrieve the wheelchair, the robot must be able to both detect it, and determine where to grab it. Our method will help the robot to get close to the grabbing position, after which a detailed analysis of scene geometry in a small region of interest can establish the grasp [17]. We divide our experiment in two parts. First, we quantitatively evaluate the resulting annotations with a large set of controlled images. Next, we evaluate the recognition ability with a set of challenging real-world images.

We collected 141 images of wheelchairs from Google Image Search. We chose semi-profile views because they were the most widely available. Note that while the ISM system can only handle a single pose, it can be extended to handle multiple viewpoints [21]. All images were annotated with ground truth part segmentations for grab area, wheels, armrests, seat, and frame. The grab area is the most important for this experiment. A few representative images and their ground truth annotations can be seen in the left and middle rows of Fig. 4.

The images are randomly split into a training and test set. We train an ISM system using 80 images, using a Hessian-Laplace interest point detector [13] and Shape Context descriptors [2]. Next, we test the system on the remaining 61 images, using the method from Section III-A. Because each image only contains one object, we select the detection with highest score for meta-data transfer. Some of the resulting annotations can be seen in the third row of Fig. 4. The grab area is found quite precisely.

To evaluate this experiment quantitatively, we use the ground truth annotations to calculate the following error measures. We define *leakage* as the percentage of background

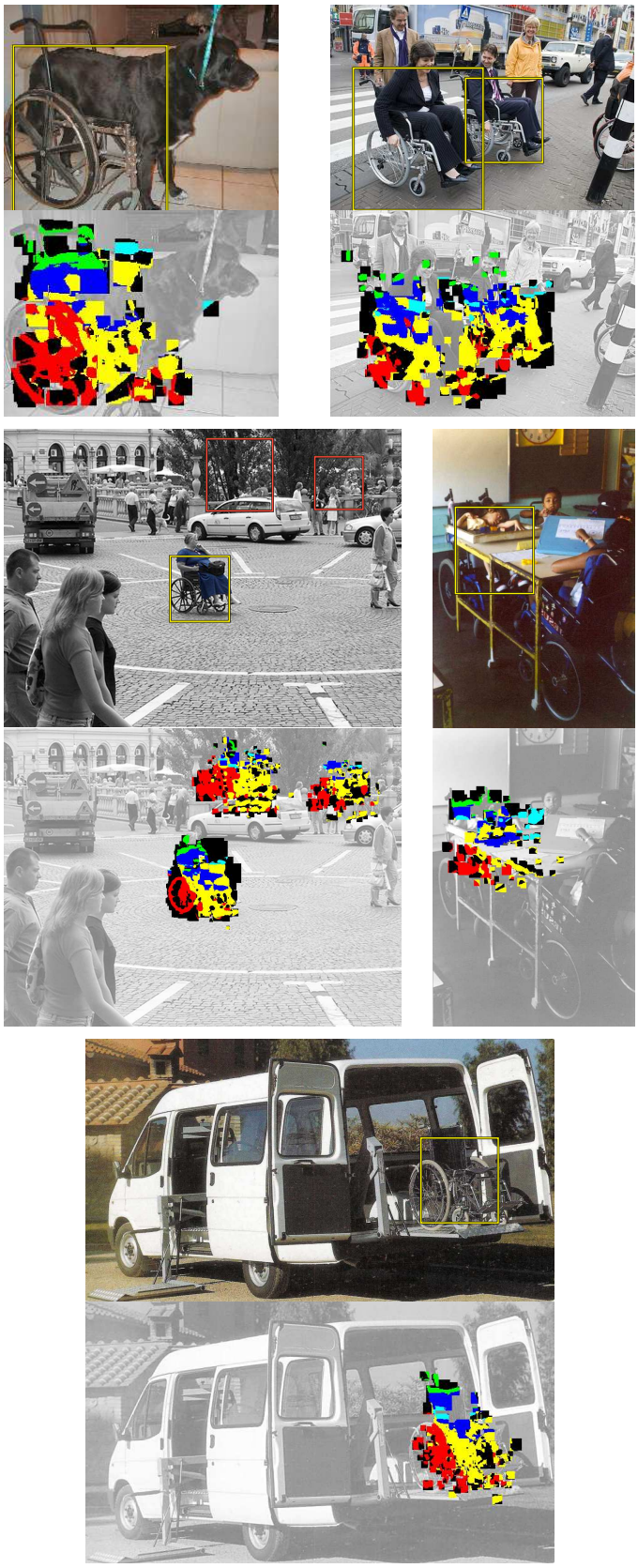

Fig. 5. Wheelchair detection and annotation results on challenging real-world test images (best viewed in color). Yellow and red rectangles indicate correct and false detections respectively. Note how one wheelchair in the middle right image was missed because it is not in the pose used for training.

	backgrnd	frame	seat	armrest	wheels	**grab-area**	unlabeled
backgrnd	**32.58**	1.90	0.24	0.14	1.10	0.37	**63.67**
frame	15.29	**66.68**	6.47	0.46	6.90	0.10	4.10
seat	2.17	15.95	**74.28**	0.97	0.33	1.55	4.75
armrest	11.22	5.62	29.64	**49.32**	1.25	0.63	2.32
wheels	13.06	9.45	0.36	0.07	**71.39**	0.00	5.67
grab-area	6.48	1.28	9.77	0.11	0.00	**76.75**	5.62

TABLE I

CONFUSION MATRIX FOR THE WHEELCHAIR PART ANNOTATION EXPERIMENT. THE ROWS REPRESENT THE ANNOTATION PARTS IN THE GROUND-TRUTH MAPS, THE COLUMNS THE OUTPUT OF OUR SYSTEM. THE LAST COLUMN SHOWS HOW MUCH OF EACH CLASS WAS LEFT UNLABELED. FOR MOST EVALUATIONS, THOSE AREAS CAN BE CONSIDERED 'BACKGROUND'.

pixels in the ground-truth annotation that were labeled as non-background by the system. The leakage for this experiment, averaged over all test images, is 3.75%. We also define a *coverage* measure, as the percentage of non-background pixels in the ground-truth images labeled non-background by the system. The coverage obtained by our algorithm is 95.1%. This means our method is able to reliably segment the chair from the background.

We evaluate the annotation quality of the separate parts with a confusion matrix. For each image, we count how many pixels of each part a_j in the ground-truth image are labeled by our system as each of the possible parts (grab, wheels, etc.), or remain unlabeled (which can be considered background in most cases). This score is normalized by the total number of pixels in the ground-truth $\hat{a}_j$. We average the confusion table entries over all images, resulting in Table I. The diagonal elements show how well each part was recovered in the test images. Not considering the armrests, the system performs well as it labels correctly between 67% and 77% of the pixels, with the latter score being for the part we are the most interested in, i.e. the grab area. The lower performance for the armrests is due to the fact that it is the smallest part in most of the images. Small parts have higher risk of being confused with the larger parts in their neighborhood.

To test the detection ability of our system, we collected a set of 34 challenging real-world images with considerable clutter and/or occlusion. We used the same ISM system as in the annotation experiment, to detect and annotate the chairs in these images. Some results are shown in Fig. 5. We consider a detection to be correct when its bounding box covers the chair. Out of the 39 wheelchairs present in the images, 30 were detected, and there were 7 false positives. This corresponds to a recall of 77% and a precision of 81%.

B. Cars: Optimizing an Automated Car Wash

In further experiments, we infer different types of meta-data for the object class 'car'. In the first experiment, we decompose recognized cars in their most important parts, similarly to the wheelchairs. In the second experiment, approximate 3D information is inferred. A possible application is an automated car wash. As illustrated in Fig. 1, the decomposition in parts can be used to apply different washing methods to the

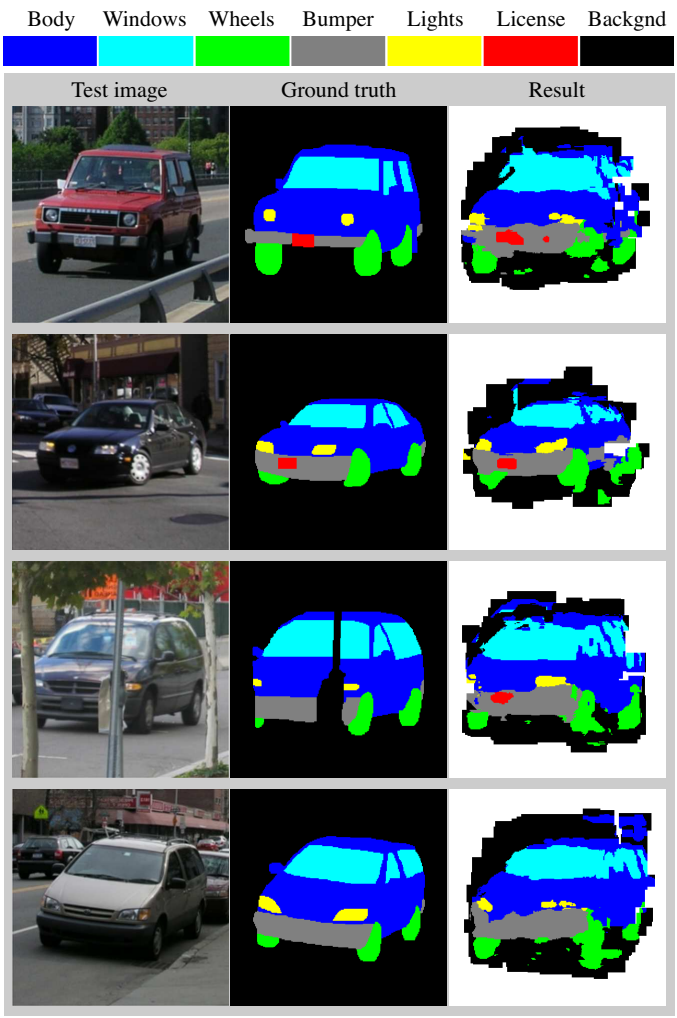

Fig. 6. Results for the car annotation experiment. From left to right: test image, ground-truth, and output of our system. White areas are unlabeled and can be considered background.

different parts. Moreover, even though such systems mostly have sensors to measure distances to the car, they are only used locally while the machine is already running. It could be useful to optimize the washing process beforehand, based on the car's global shape inferred by our system.

Our dataset is a subset of that used in [9]. It was obtained from the LabelMe website [16], by extracting images labeled as 'car' and sorting them according to their pose. For our experiments, we only use the 'az300deg' pose, which is a semi-profile view. In this pose both the front (windscreen, headlights, license plate) and side (wheels, windows) are visible. This allows for more interesting depth maps and part annotations compared to pure frontal or side views. The dataset contains a total of 139 images. We randomly picked 79 for training, and 60 for testing.

For parts annotation, the training and testing phase is analogous to the wheelchair experiment (section IV-A). Results are shown in Fig. 6. The leakage is 6.83% and coverage is 95.2%. The confusion matrix is shown in Table II. It again

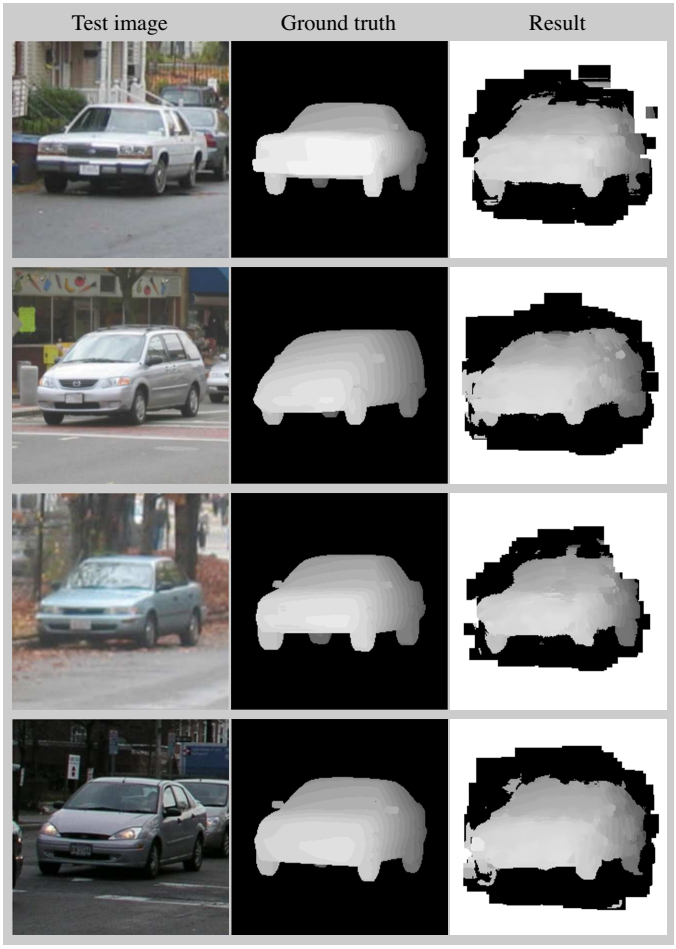

Fig. 7. Results for the car depthmap experiment. From left to right: test image, ground-truth, and output of our system. White areas are unlabeled and can be considered background.

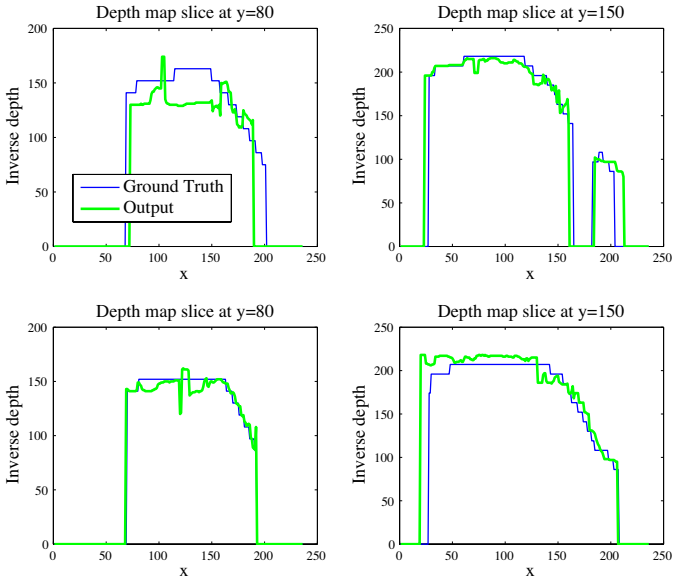

Fig. 8. Horizontal slices through the ground truth and output depthmaps of the second car (top row) and fourth car (bottom row) in Fig. 7.

	bkgnd	body	bumper	headlt	window	wheels	license	unlabeled
bkgnd	**23.56**	2.49	1.03	0.14	1.25	1.88	0.04	**69.61**
body	4.47	**72.15**	4.64	1.81	8.78	1.86	0.24	6.05
bumper	7.20	4.54	**73.76**	1.57	0.00	7.85	2.43	2.64
headlt	1.51	36.90	23.54	**34.75**	0.01	0.65	0.23	2.41
window	3.15	13.55	0.00	0.00	**80.47**	0.00	0.00	2.82
wheels	11.38	6.85	8.51	0.00	0.00	**63.59**	0.01	9.65
license	2.57	1.07	39.07	0.00	0.00	1.04	**56.25**	0.00

TABLE II

CONFUSION MATRIX FOR THE CAR PARTS ANNOTATION EXPERIMENT (CFR. TABLE I).

shows good labeling performance, except for the headlights. Similarly to the armrests in the wheelchair experiments, this is as expected. The headlights are mostly very small, hence easily confused with the larger parts (body, bumper) in which they are embedded.

For the depth map experiment, we obtained ground-truth data by manually aligning the best matching 3D model from a freely available collection[2] to each image, and extracting the OpenGL Z-buffer. In general, any 3D scanner or active lighting setup could be used to automatically obtain depth maps. We normalize the depths based on the dimensions of the 3D models, by assuming that the width of a car is approximately constant. The depth maps are quantized to 20 discrete values. Using these maps as annotations, we use our method of section III-B to infer depths for the test images.

Results are shown in the rightmost column of Fig. 7. The leakage is 4.79% and the coverage 94.6%, hence the segmentation performance is again very good. It is possible to calculate a real-world depth error estimate, by scaling the normalized depth maps by a factor based on the average width of a real car, which we found to be approximately 1.8m. All depth maps are scaled to the interval $[0, 1]$ such that their depth range is 3.5 times the width of the car, and the average depth error is 0.042. This is only measured inside areas which are labeled non-background in both the ground-truth and result images, to eliminate bias from the background. A plausible real-world depth error can therefore be calculated by multiplying this figure by $3.5 \cdot 1.8$m, which yields a distance of 27cm. To better visualize how the output compares to the ground truth, Fig. 8 shows a few horizontal slices through two depth maps of Fig. 7.

To illustrate the combined recognition and annotation ability of our system for this object class, we again tested it on real-world images. We used the same system as in the annotation experiment on a few challenging images containing cars in a similar pose, including the car wash image from Fig. 1. Results are shown in Fig. 9.

V. CONCLUSIONS

We have developed a method to transfer meta-data annotations from training images to test images containing previously unseen objects, based on object class recognition. Instead of using extra processing for the inference of meta-data, it

[2]http://dmi.chez-alice.fr/models1.html

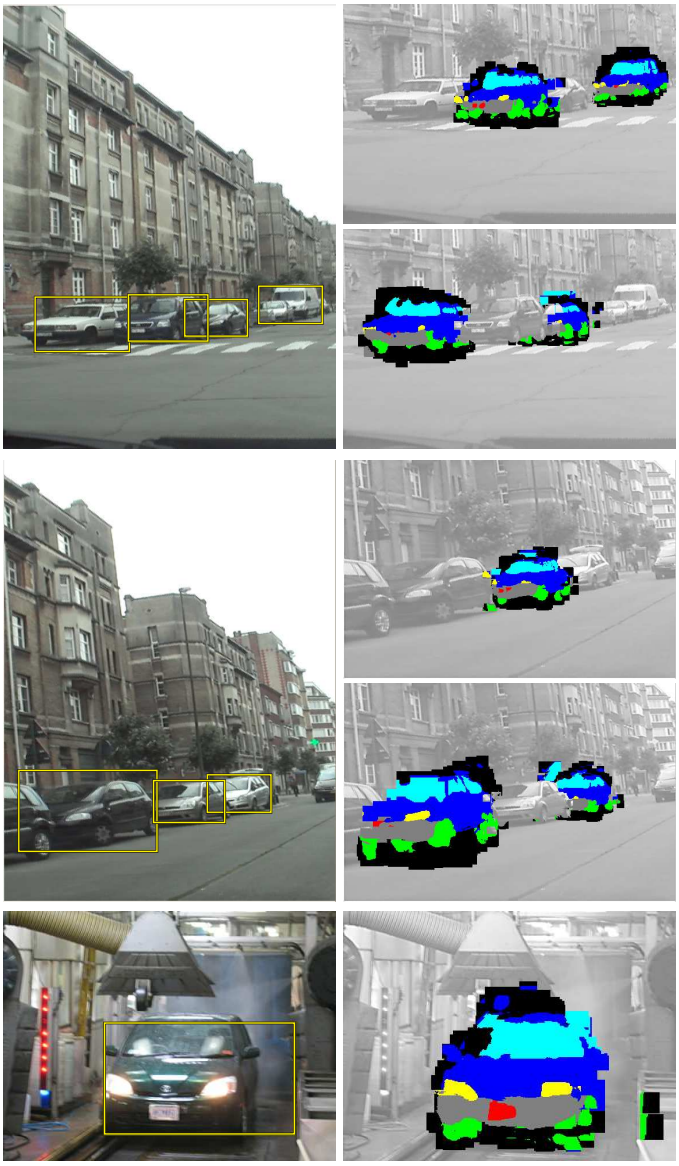

Fig. 9. Car detection and annotation results on real-world test images. Even though the car from Fig. 1 is in a near-frontal pose, it was still correctly detected and annotated by the system trained on semi-profile views.

is deeply intertwined with the actual recognition process. Low-level cues in an image can lead to the detection of an object, and the detection of the object itself causes a better understanding of the low-level cues from which it originated. The resulting meta-data inferred from the recognition can be used to initiate or refine robot actions.

Future research includes using the output from our system in a real-world application to guide a robot's actions, possibly in combination with other systems. We will also investigate methods to improve the quality of the annotations by means of relaxation or Markov Random Fields, and ways to greatly reduce the amount of manual annotation work required for training.

Acknowledgment

The authors gratefully acknowledge support by IWT-Flanders, Fund for Scientific Research Flanders and European Project CLASS (3E060206).

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Proceedings ECCV, Springer LNCS*, 3951(1):404–417, 2006.

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000.

[3] F. Fraundorfer, C. Engels, and D. Nister. Topological mapping, localization and navigation using image collections. *IROS*, 2007.

[4] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool. Omnidirectional vision based topological navigation. *Int. Journal on Computer Vision and Int. Journal on Robotics Research*, 74(3):219–236, 2007.

[5] F. Han and S.-C. Zhu. Bayesian reconstruction of 3D shapes and scenes from a single image. *Workshop Higher-Level Knowledge in 3D Modeling Motion Analysis*, 2003.

[6] T. Hassner and R. Basri. Example based 3D reconstruction from single 2d images. In *Beyond Patches Workshop at IEEE CVPR06*, page 15, June 2006.

[7] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *CVPR*, pages 2137–2144, 2006.

[8] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. *ICCV*, 2005.

[9] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3D scene analysis from a moving vehicle. *CVPR*, 2007.

[10] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. *BMVC*, 2003.

[11] B. Leibe and B. Schiele. Pedestrian detection in crowded scenes. *CVPR*, 2005.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[14] D. Mumford. Neuronal architectures for pattern-theoretic problems. *Large-Scale Neuronal Theories of the Brain*, pages 125–152, 1994.

[15] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 66(3):231–259, 2006.

[16] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. LabelMe: a database and web-based tool for image annotation. *MIT AI Lab Memo AIM-2005-025*, 2005.

[17] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng. Robotic grasping of novel objects. *NIPS 19*, 2006.

[18] A. Saxena, J. Schulte, and A. Y. Ng. Learning depth from single monocular images. *NIPS 18*, 2005.

[19] Sinisa Segvic, Anthony Remazeilles, Albert Diosi, and Francois Chaumette. Large scale vision-based navigation without an accurate global reconstruction. *CVPR*, 2007.

[20] E.B. Sudderth, A. Torralba, W.T. Freeman, and A.S. Willsky. Depth from familiar objects: A hierarchical model for 3D scenes. *CVPR*, 2006.

[21] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. *CVPR*, 2006.